Image and Vision Computing 60 (2017) 38–47

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Semi-supervised learning through adaptive Laplacian graph trimming ☆

Zongsheng Yue[a], Deyu Meng[a,b,*], Juan He[a], Gemeng Zhang[a]

[a] School of Mathematics and Statistics and Ministry of Education Key Lab of Intelligent Networks and Network Security, Xi'an Jiaotong University, Shaanxi, PR China
[b] Faculty of Information Technology, Macau University of Science and Technology, Macau, PR China

ARTICLE INFO

Article history:
Received 16 May 2016
Received in revised form 5 September 2016
Accepted 16 November 2016
Available online 24 November 2016

Keywords:
Semi-supervised learning
Graph Laplacian
Self-paced learning
Nearest neighborhood graph

ABSTRACT

Graph-based semi-supervised learning (GSSL) attracts considerable attention in recent years. The performance of a general GSSL method relies on the quality of Laplacian weighted graph (LWR) composed of the similarity imposed on input examples. A key for constructing an effective LWR is on the proper selection of the neighborhood size K or ε on the construction of KNN graph or ε-neighbor graph on training samples, which constitutes the fundamental elements in LWR. Specifically, too large K or ε will result in "shortcut" phenomenon while too small ones cannot guarantee to represent a complete manifold structure underlying data. To this issue, this study attempts to propose a method, called adaptive Laplacian graph trimming (ALGT), to make an automatic tuning to cut improper inter-cluster shortcut edges while enhance the connection between intra-cluster samples, so as to adaptively fit a proper LWR from data. The superiority of the proposed method is substantiated by experimental results implemented on synthetic and UCI data sets.

1. Introduction

Semi-supervised learning (SSL) aims at solving the common situation where labeled data are scarce but unlabeled data are abundant. Under suitable assumptions, it uses unlabeled data to help supervised learning tasks. Various SSL methods have been proposed and shown promising results in recent decades, such as self-training [25], mixture models [5,24] and graph-based semi-supervised learning (GSSL) [16,30]. Thereinto, the GSSL approach has been attracting increasing attention in many real applications, including sentiment categorization [8], image colourisation [15], holistic object categorization [7], because of its fine out-of-sample prediction capability, low cost of calculation and high degree of accuracy.

The current developments of GSSL methods can be represented as the following typical works. The mincut method [4] converts the SSL to the problem of seeking a minimum cut in a graph. The Gauss random fields and harmonic function method [34] and the global and consistency method [33], handle the label prediction by making possibly optimal trade-off between the accuracy of the classifier on the labeled data and a regularizer term which reflects the smoothness condition on the entire data. On the basis of how easy the random walk goes from one example to another, Szummer [26] proposed the Markov random walk method. Besides, the manifold regularization method [3,22] considers SSL in a new framework for data-dependent regularization and exploits the geometry of the probability of the distribution framework. This method not only has elegant theories but also inclines to achieve promising performance under certain complicated data set [19]. Recently, a bivariate framework with an efficient solution via greedy gradient Max-Cut for GSSL method [28] is also proposed by simultaneously optimizing the binary label information and a continuous classification function.

These GSSL methods need to pre-specify a nearest neighborhood (NN) graph superimposed on the entire dataset, and the K-NN or ε-NN strategy is commonly used to build point-pairwise edges within a similar class to represent the intra-class similarity knowledge. Such NN graph construction strategy [23], however, has evident limitations in real applications. Firstly, the neighborhood size of K or ε will easily influence the performance of GSSL methods [17,18]. Intuitively, the smaller K or ε inclines to be not capable of capturing enough neighborhood examples' information as shown in Fig. 1 (a), while the larger ones may possibly cause some wrong connections between different classes as the red lines depicted in Fig. 1 (b). Here we regard these unexpected connections as "shortcut", which correspond to the false inter-class neighborhood connection and tends to negatively influence of the utilized GSSL method. Secondly,

☆ This paper has been recommended for acceptance by Xi Peng, Ph.

* Corresponding author at: School of Mathematics and Statistics and Ministry of Education Key Lab of Intelligent Networks and Network Security, Xi'an Jiaotong University, Shaanxi, PR China.
E-mail address: dymeng@mail.xjtu.edu.cn (D. Meng).

http://dx.doi.org/10.1016/j.imavis.2016.11.013
0262-8856/

the weight of each neighborhood connection/edge is generally subjectively evaluated. Provided that these weights have been set before the GSSL run, they have no chance to be adapted any more by the feedback of the classification effects in the learning process. That means that if we have mistakenly involved some false weights, especially for those between inter-class nearest neighboring samples, before the implementation of GSSL, they have to be consistently utilized and cannot be rectified in later training. To address the above issues, we propose a novel robust method, called adaptive Laplacian graph trimming (ALGT) SSL method, which is capable of automatically fitting a proper Laplacian matrix (i.e., weights between all NN samples) and less sensitive to the choice of the neighborhood size. Specifically, even under a relatively large neighborhood scale, the method can adaptively break the inter-class shortcuts while preserve the intra-class NN connection between samples. The performance of the method can thus be evidently ameliorated beyond previous state-of-the-art methods along this line.

The paper is organized as follows. In Section 2, we briefly review some related works. Then the proposed ALGT-SSL problem as well as its solving strategy is introduced in Section 3. Section 4 presents the experimental results, and in Section 5 we make concluding remarks and discussions.

2. Background and related work

Assume that $P(X,Z)$ is the probability distribution between data and label variables, and the training data are independent and identically distributed (i.i.d.) samples $\{(x_1,z_1),(x_2,z_2),\cdots,(x_l,z_l)\}$ drawn from $P(X,Z)$ and unlabeled samples $\{x_{l+1},x_{l+2},\cdots,x_{l+u}\}$ are generated from the marginal distribution $P(X)$. For SSL, we always consider $l \ll n$, where $n = l+u$. The aim of SSL is to infer the labels $\{z_{l+1},z_{l+2},\cdots,z_{l+u}\}$ for those unlabeled samples.

The first step of a GSSL method is to construct the adjacency graph $G = \{X,E\}$, where $X=\{x_1,x_2,\cdots,x_n\}$ is the vertex set and $E = \{e_{ij}\}$ stands for the edges set. Then we can define weighted matrix $W = \{w_{ij}\}$ and degree matrix

$$D = diag\{d_1,d_2,\cdots,d_n\}$$

where w_{ij} represents the weight of edge e_{ij}, namely, the similarity between vertexes x_i and x_j, and $d_i = \sum_{j=1}^{n} w_{ij}$. Then the graph Laplacian is defined as $L = D - W$.

2.1. Weighted graph construction

Given n samples $X=\{x_1,x_2,\cdots,x_n\}$, the weighted graph construction usually contains two processes [1,28].

1. Constructing NN-graph
   There are two main strategies to determine whether to connect vertices x_i and x_j in the NN-graph imposed on the training data.
   (1) ε-NN. ε is regarded as a threshold for the distance between node i and node j. It means that we connect vertices x_i and x_j if $\|x_i - x_j\|^2 < \varepsilon$, where $\|\cdot\|$ is usually Euclidean distance or cosine distance.
   (2) K-NN. We put an edge to connect them if node i is one of the K nearest neighbors of j or vice versa.
   Both methods have their pros and cons. The ε-NN strategy can generate a symmetric relationship, while the K-NN cannot promise this fact. However, the appropriate threshold ε is relatively harder to select and often lead to disconnected graphs. In comparison, the selection of K is relatively easier and convenient. Hence, we prefer to use K-NN in all our experiments.
2. Defining edge weights of the NN-graph.
   There are two commonly utilized strategies to define weights for all edges of the NN-graph.
   (1) Gaussian Kernel weight:

$$w_{ij} = \begin{cases} exp(-\frac{\|x_i - x_j\|^2}{2t^2}), & \textbf{if } \textit{nodes i and j are connected} \\ 0, & \textit{otherwise} \end{cases} \tag{1}$$

   where t is bandwidth parameter.
   (2) Binary Weight.

$$w_{ij} = \begin{cases} 1, & \textbf{if } \textit{nodes i and j are connected}, \\ 0, & \textit{otherwise}. \end{cases} \tag{2}$$

2.2. Manifold regularization

We then introduce the main idea of the manifold regularization method [3,22], and then propose our ALGT-SSL method based on it. ALGT variations on other GSSL methods, such as LGC [33] and GFHF [34], can be easily extended.

Unlabeled data are always useful to provide us useful prior information under the hypothesis that $P(X)$ is in relation to the conditional probability $P(Z|X)$. In other words, if one sample x_i is quite adjacent to another sample x_j on the intrinsic geometry of $P(X)$, then their corresponding labels z_i and z_j should be possibly similar. When the support of $P(X)$ is a compact sub-manifold $\mathcal{M} \subset X = R^n$, after constructing a NN-graph on data, this assumption can be encoded as a regularization term using the graph Laplacian L as following [2,3,22]:

$$\begin{aligned} f^* &= \underset{f\in\mathcal{H}_K}{\operatorname{argmin}} \frac{1}{l}\sum_{i=1}^{l} L(x_i,y_i,f) + \gamma_A \|f\|_K^2 + \gamma_I \hat{f}^T L \hat{f} \\ &= \underset{f\in\mathcal{H}_K}{\operatorname{argmin}} \frac{1}{l}\sum_{i=1}^{l} L(x_i,y_i,f) + \gamma_A \|f\|_K^2 + \gamma_I \sum_{i,j} w_{ij}(\hat{f}_i - \hat{f}_j)^2, \end{aligned} \tag{3}$$

where $L(\cdot)$ is certain loss function, $\mathcal{H}_K$ is an appropriately chosen Reproducing Kernel Hilbert Space (RKHS) and $\|\cdot\|_K$ is the corresponding norm of this RKHS which measures the smoothness degree on the outputted decision values of all samples along the underlying data manifold, and $\hat{f} = (\hat{f}_1,\hat{f}_2,\cdots,\hat{f}_n)$. Here γ_A controls the complexity of the function in the ambient space while γ_I compromises the complexity of the function in the intrinsic geometry of $P(X)$. Specifically, if $L(x_i,y_i,f) = (y_i - f(x_i))^2$ or $L(x_i,y_i,f) = \max\{0,1-y_i f(x_i)\}$, we can get the forms of Laplacian Regularization Least Squares (LapRLS) or Laplacian Support Vector Machine (LapSVM) [3,22], respectively.

The regularization term $\hat{f}^T L \hat{f} = \frac{1}{2}\sum_{i,j} w_{ij}(\hat{f}_i - \hat{f}_j)^2$ ensures the establishment of the hypothesis as mentioned above. According to Eq. (3), the effect of minimizing enforces $\hat{f}_i$ and $\hat{f}_j$ possibly similar when w_{ij} is of relatively large positive value. However, as for shortcuts, the minimum effect would mistakenly make $\hat{f}_i$ and $\hat{f}_j$ similar on two samples with different categories. Thus, too large neighborhood size will raise many shortcuts, and always tends to seriously degenerate the performance of the corresponding GSSL method. So generally the traditional GSSL methods are sensitive to the selection of the neighborhood size parameter. To alleviate this robustness issue, we design our ALGT-SSL strategy as introduced in the following.

3. Adaptive Laplacian graph trimming

Constructing a rational NN-graph, representing the intra-class NN knowledge on data, intrinsically affects the performance of a

GSSL approach. Thus, there has been some research attempting to learn the adjacency matrix or weighted (similar) matrix, such as b-matching [10], FGV [6], adaptive KNN [21] and so on. However, performance of most of these methods depend on the presetting neighbor size K. In most applications, an appropriate K is difficult to chose. What's more, once their values are set, we cannot adaptively ameliorate them by virtue of some useful feedbacks from the later learning process any more (e.g., two samples are with the same or different classes). Hence, this study attempts to initially provide some new insights on both issues.

Specifically, we present the following ALGT-SSL framework based on manifold regularization. This new strategy has following properties:

(1) For fixed K or ε, the weight w_{ij} can be gradually rectified based on the feedback of label information in the learning process of a GSSL method to break up inter-class edges while enhance intra-class ones of the NN-graph.
(2) The configuration of the NN-graph can also be adaptively ameliorated in this learning process, instead of pre-fixed as conventional.
(3) Our method is less sensitive to the choice of neighborhood size because of its self-rectified NN-structures.

Since our method is inspired by a newly emerging machine learning approach called self-paced learning (SPL), we first make a short review for it.

3.1. Self-paced learning

SPL [13,14,20] has attracted increasing attention in the field of machine learning [11,12,31] and computer vision [9,27,29] since it was raised in 2010. It aims to calculate the following problem:

$$\min_{w,v} \mathbb{E}(w,v;\lambda) = \sum_{i=1}^{n} v_i L(y_i, g(x_i,w)) + f(v;\lambda), s.t. v \in [0,1]^n,$$

where $L(\cdot)$ is certain loss function that measures the cost between the ground truth label y_i and the estimated one $g(x_i,w)$, $v = [v_1, v_2, \cdots v_n]^T$ denote the weight variables reflecting the samples' importance, λ represents "age" parameter controlling the learning pace, $f(v;\lambda)$ is the so called self-paced regularizer, which is defined by Jiang et al. and Zhao et al. [13,32] as follows:

Definition self-paced regularizer. Suppose that $v = [v_1, v_2, \cdots v_n]^T$ is a vector of weight variable for each training sample and $l = [l_1, l_2, \cdots, l_n]^T$ is the corresponding loss, and λ is the "age" parameter controlling the learning pace. $f(v;\lambda)$ is called self-paced regularizer, if

1. $f(v;\lambda)$ is convex with respect to $v \in [0,1]^n$.
2. When all variables are fixed except for v_i, l_i, v_i^* decreases with l_i, and it holds that $\lim_{l_i \to 0} v_i^* = 1$, $\lim_{l_i \to \infty} v_i^* = 0$.
3. v_i^* monotonically increases with respect to λ, and it holds that for $\forall i \in \{1,2,\cdots,n\}$, $\lim_{\lambda \to \infty} v_i^* = 1$, $\lim_{\lambda \to 0} v_i^* = 0$.

where

$$v^* = \arg\min_{v \in [0,1]^n} \sum_{i=1}^{n} v_i l_i + f(v;\lambda)$$

Condition 2 indicates that the model emphasizes those samples with smaller losses in learning, and Condition 3 states that when the model "age" λ gets larger, it tends to incorporate more, probably complex samples into training.

It is easy to see that if we substitute sample-pairwise distances d_{ij} to the losses in the SPL model, then the optimal weight w_{ij} satisfy that it is monotonically decreasing w.r.t. d_{ij} (Condition 2), and

$$\lim_{d_{ij} \to \infty} w_{ij} = 0, \lim_{d_{ij} \to 0} w_{ij} = 1.$$

It is interesting that such insight exactly complies with the weight valuing strategy in the NN-graph. Specifically, weights between closer samples should have large weights than those farther ones. We thus expect to employ such learning strategy to design an adaptively NN-structure learning strategy for a general GSSL method.

3.2. ALGT model

Based on the aforementioned, we proposed the ALGT-GSSL model as follows:

$$\min_{f,W} \mathbf{E}(W,f) = \frac{1}{l}\sum_{i=1}^{l} L_i(x_i,y_i,f_i) + \gamma_A \|f\|_K^2 + \gamma_I \hat{f}^T L \hat{f} + \gamma_X tr(X^T L X) + f(W;\lambda) \quad (4)$$

$$s.t. w_{i,\delta_i(j)} = 0, if \quad j > K, i = 1,2,\cdots,n$$

where $\delta_i(j)$ represents the index of the ascending order of d_{ij} for $i = 1,2,\cdots,n$, and K is the neighborhood size, γ_X is regularization parameter that controls to what extent does the similarity of two points influence the weighted matrix of the graph, $f(W;\lambda)$ represents regularizer term, λ is the "age" parameter, also functioning in the neighborhood size (details are analyzed in the following), $\hat{f} = (\hat{f}_1, \hat{f}_2, \cdots, \hat{f}_n)$, and the other symbols has the same meaning as Eq. (3).

For the GSSL context, we utilize two specific forms of $f(W;\lambda)$ listed following:

$$f(W;\lambda) = -\lambda \sum_{i,j} w_{ij}, \quad (5)$$

and

$$f(W;\lambda) = \lambda \sum_{i,j} (w_{ij} \ln w_{ij} - w_{ij}). \quad (6)$$

By utilizing these two self-paced regularizer and adopting $L_i(x_i,y_i,f)$ as squared loss function $(y_i - f(x_i))^2$ or hinge loss function $max[0, 1 - y_i f(x_i)]$ in Eq. (4), we can get to-be-investigated ALGT Regularized Least Square (ALGTLRLS) or ALGT Support Vector Meachine (ALGTSVM) problems.

More insights on the model can be more conveniently interpreted together with introducing the details of the solving strategy for it as follows.

3.3. ALGT algorithm

ACS (Alternative Convex Search) is a generally used strategy to solve biconvex optimization problems [14], in which the variables in the problem are divided into multiple disjoint blocks. In every iteration, a block of variables are optimized while keeping the other blocks fixed. Inspired by ACS and the algorithm in [13,32], an ACS algorithm is proposed to solve Eq. (4). The details are summarized in Algorithm 1. The algorithm takes inputs of labeled and unlabeled samples, and outputs an optimal weighted graph and a decision function. It iterates alternately for the following two steps until it finally converges: Step 3 learns the optimal decision function f^* with fixed weighted graph W, and step 4 learns the optimal weighted graph W^* with fixed decision function f. In step 3, we can solve the problem using the conventional LapSVM algorithm [3,19]. And in each iteration of step 4, it rectifies a

better weighted graph automatically based on the feedback of label information. Details are introduced as follows:

Algorithm 1. ALGT semi-supervised learning

ALGT semi-supervised learning

Input: Labeled and unlabeled samples
Output: W^*, f^*
1: Initialize $W^* = W_0$
2: **while** not converged **do**
3: Update $f^* = argmin_f \mathbf{E}(W^*, f)$
4: Update $W^* = argmin_f \mathbf{E}(W, f^*)$
5: **end while**
6: **return** W^*, f^*

Under fixed f, the optimal weights W^* can be calculated in closed-form. When we adopt regularizer as Eq. (5), it is easy to calculate the solution of W^* as:

$$w_{ij}^* = \begin{cases} 1 & \textbf{if} \quad d_{ij} < \lambda \\ 0 & else. \end{cases} \tag{7}$$

And for another regularizer Eq. (6), the solution has the form:

$$w_{ij}^* = \begin{cases} exp(-\frac{d_{ij}}{\lambda}) & \textbf{if} \quad d_{ij} < \lambda \\ 0 & else \end{cases} \tag{8}$$

where

$$d_{ij} = \gamma_I(\widehat{f}_i - \widehat{f}_j)^2 + \gamma_X \|x_i - x_j\|^2. \tag{9}$$

It is easy to see that this updated weights integrated the feedback information on the current label prediction through $(\widehat{f}_i - \widehat{f}_j)^2$ besides the traditional distance information. If d_{ij} is larger than the threshold λ, the edge between vertices x_i and x_j will be broken up (i.e., $w_{ij} = 0$), otherwise they will be connected (i.e., $w_{ij} \neq 0$). Beyond the traditional weight specification strategy for NN-graph, some sample pairs located in different classes incline to more possibly separated even they are close to each other. This brings the main insightful capability of the presented method.

It should be noted that the difference between Eq. (7) and Eq. (8) is the definition of weight matrix: the former implements a "hard" weighting, which means that the weight only equals to 0 or 1, while

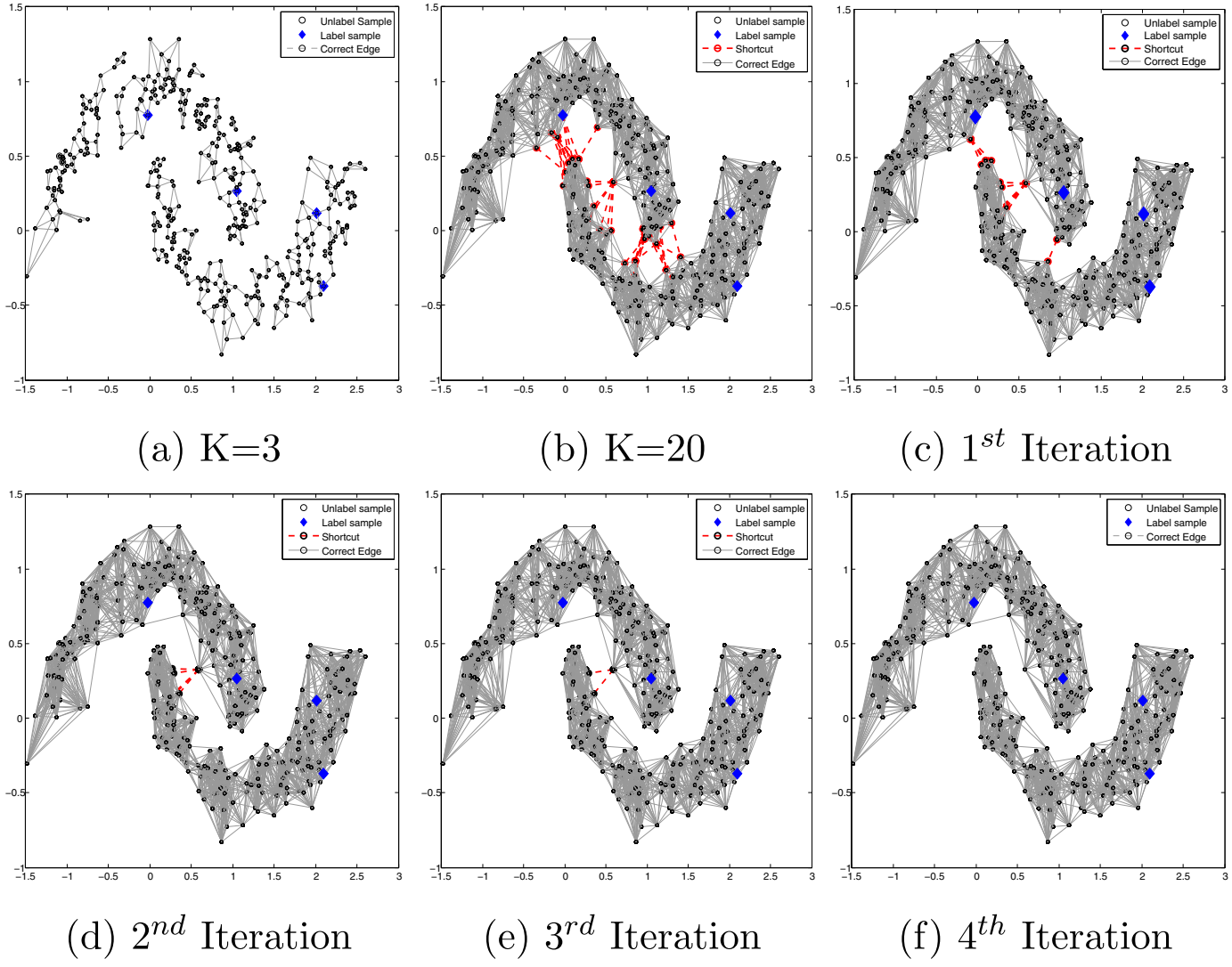

Fig. 1. (a) and (b) NN-graphs of the two moons data set under neighborhood sizes 3 and 20; The shortcut phenomenon in (b) can be easily observed. (c), (d), (e) and (f) NN-graphs obtained by the proposed ALGTSVM method in different iteration numbers under initial neighborhood size 20. It is seen that all shortcuts are gradually broken up during the training process of the method.

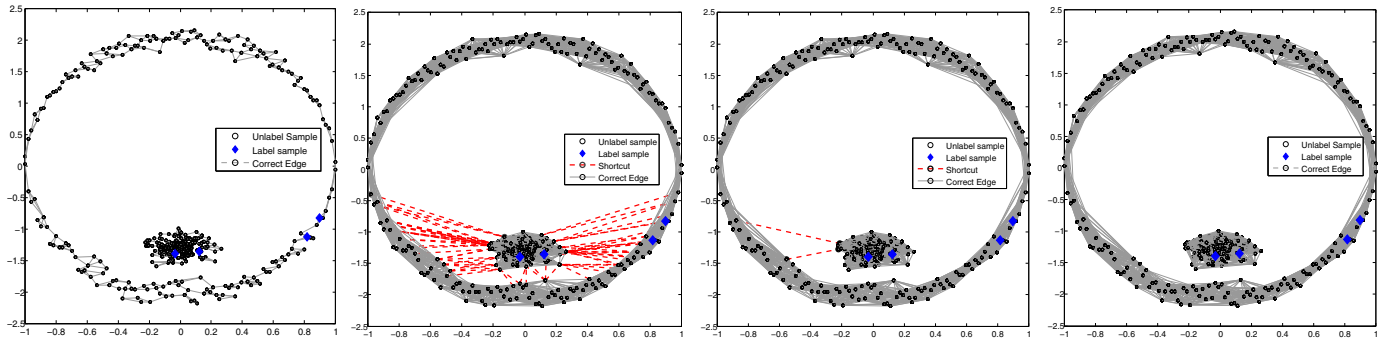

Fig. 2. (a) and (b) NN-graphs of the ellipse data set under neighborhood sizes 3 and 20; (c) and (d) NN-graphs obtained by the proposed ALGTSVM method in different iteration numbers under initial neighborhood size 20.

the latter leads to "soft" weighting strategy that assigns real-valued weights between 0 and 1 to samples.

It should be indicated that the traditional weight specification strategy for a NN-graph can be taken as a special case of our model. If we set $\gamma_X = 1$ and $\gamma_I = 0$ in Eq. (9), the optimal weight matrix in Eq. (7) and Eq. (8) is exactly the same as the previous binary weight Eq. (2) and Gauss kernel weight Eq. (1), respectively, as introduced in Section 2. Especially, if we initialize W_0 as introduced in Section 2, then in the next step the updating of $f = argmin_f \mathbf{E}(W_0, f)$ is exactly equivalent to the manifold regularization GSSL method. This implies that our method actually corresponds to a gradually amelioration strategy in the fundament of the previous manifold regularization GSSL method.

Actually, the proposed ALGT-SSL can be generally adopted to ameliorate the capability of previous manifold regularization GSSL to involve the NN graph weight matrix into the learning process. It facilitates considering the label feedback $\left\|\widehat{f}_i - \widehat{f}_j\right\|^2$ after each iteration in learning, which leads to a self-adaptive graph trimming mechanism. Since the shortcut edges should be scattered across different classes, the corresponding value of $\left\|\widehat{f}_i - \widehat{f}_j\right\|^2$ should grow lager during learning process, which helps to break edges between inter-class samples even that they are very close in the space. The improperly specified NN-edges thus have another chance to be further rectified, which naturally makes our method more robust to neighborhood size specification.

3.4. Discussion

3.4.1. Manual annotation

For SSL, labeled examples play a key role during the learning process, which provides the most fundamental and significant information to train a classifier. However, if the labeled samples transmit insufficient information to the classifier, e.g., most labeled samples are too close to be distinguished, performance of most SSL methods, as well as the proposed one, tend to be degenerated. We thus more prefer the labeled samples possibly scattered.

Therefore, it is necessary to select appropriate labeled examples before training the model. In our experiments, we utilize Algorithm 2 to select samples which have labels. It is easy to see that this easy strategy facilitates selected samples for labels not neighboring among each other. Note that when we use this method for selecting less samples for labeling, the neighborhood size k can be set relatively larger to conduct a more scattered sample selection.

Algorithm 2. SSL Sample Selection

SSL Sample Selection

Input: Input data X (with size n), $l < n$, nearest neighbor size k
Output: To be labeled sample set X_{label}

1: Set a weighted matrix $W = \{w_{ij}\}_{n\times n}$, whose entry w_{ij} is 1 if the i-th sample is a neighborhood of the j-th one, and 0 otherwise;
2: Initialize $X_{label} = \emptyset, X_{unlabel} = X$
3: **for** $i = 1, 2, \cdots, l$ **do**
4: Randomly select a sample x_i from $X_{unlabel}$
5: $X_{label} = X_{label} \cup \boldsymbol{x_i}, X_{unlabel} = X_{unlabel}/\{x_j | w_{ij} > 0\}$
6: **end for**

3.4.2. Complexity analysis

In Algorithm 1, the most time-consuming step is to update W^* when the classifier f is fixed, i.e. step 4. Except the traditional dual theory that is similar to solve SVM [3], Melacci and Belkin put forward a simple and primal solution based on Newton's method or Conjugate Gradient, which costs no more than $O(n^3)$ [19]. Observed from Eqs. (7) and (8), the procedure of training the classifier f^* with fixed W in step 3 costs around $O(n^2)$. In our experiments, our method can get a good performance after no more than 10 iterations. The entire complexity of Algorithm 1 is thus around $O(n^3)$, which is comparable to the current state-of-the-art methods along this research line [3,19].

4. Experimental results

We present experimental results for the proposed ALGT SSL method on both synthetic and UCI data sets. The comparison method includes two state-of-the-art manifold regularization methods: LapSVM [3] and LapRLS [19]. The codes of LapSVM and LapRLS are directly unitized from the lapsvmp_v02 library[1].

In all of our experiments, part of the original data set are selected to train the classifier (based on Algorithm 2), and the remaining data are regarded as test set. We adopt the K-NN to construct the adjacency graph for all methods (also as the initial graph of our

[1] http://www.dii.unisi.it/ melacci/lapsvmp/

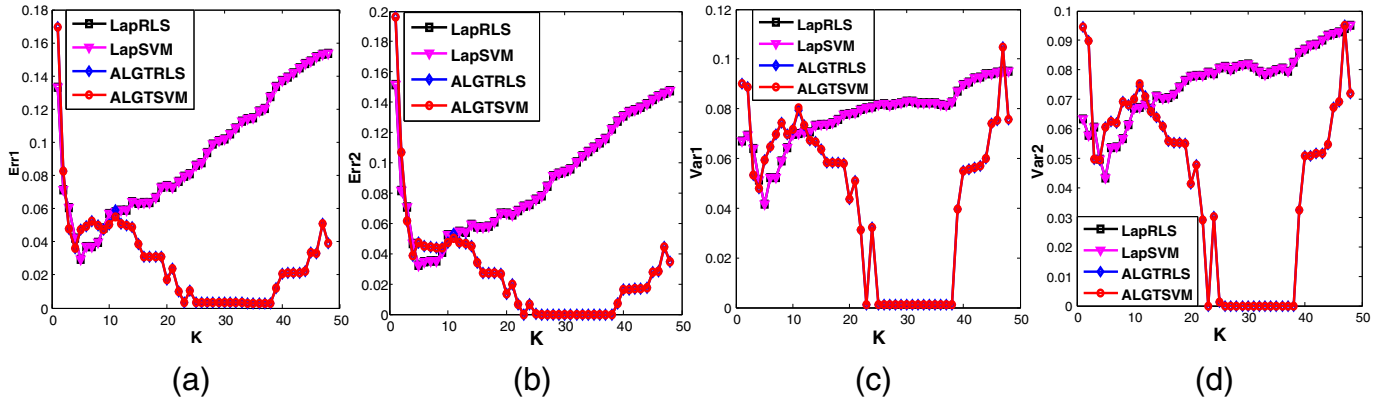

Fig. 3. Performance comparison of all competing methods on the two moons data set. Panels (a) and (b) show the predicted error rate Err1 and Err2 for unlabeled samples with respect to different neighborhood size K while panels (c) and (d) present corresponding variance.

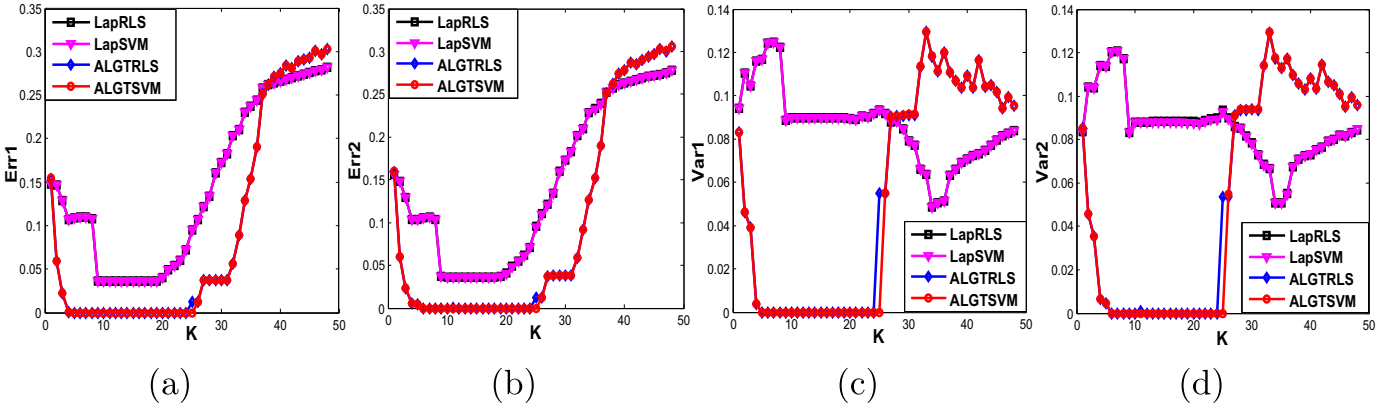

Fig. 4. Performance comparison of all competing methods on the ellipse data set. Panels (a) and (b) show the predicted error rate Err1 and Err2 for unlabeled samples with respect to different neighborhood size K while panels (c) and (d) present corresponding variance.

method), and repeated 20 independent folds of experiments. The following two error rates are utilized for performance comparison:

$$Err1 = \frac{1}{20}\sum_{i=1}^{20}\frac{\#\left\{\hat{y}_{ij}^{tr} : \hat{y}_{ij}^{tr} = y_j^{tr}, j = 1, 2, \cdots, n_1\right\}}{n_1},$$

$$Err2 = \frac{1}{20}\sum_{i=1}^{20}\frac{\#\left\{\hat{y}_{ij}^{te} : \hat{y}_{ij}^{te} = y_j^{te}, j = 1, 2, \cdots, n_2\right\}}{n_2},$$

where n_1 and n_2 represent the number of unlabeled samples within the train set and the number of samples of the test set, respectively, $\hat{y}_{ij}^{tr}$ and $\hat{y}_{ij}^{te}$ are the predicted labels of unlabeled samples in the train

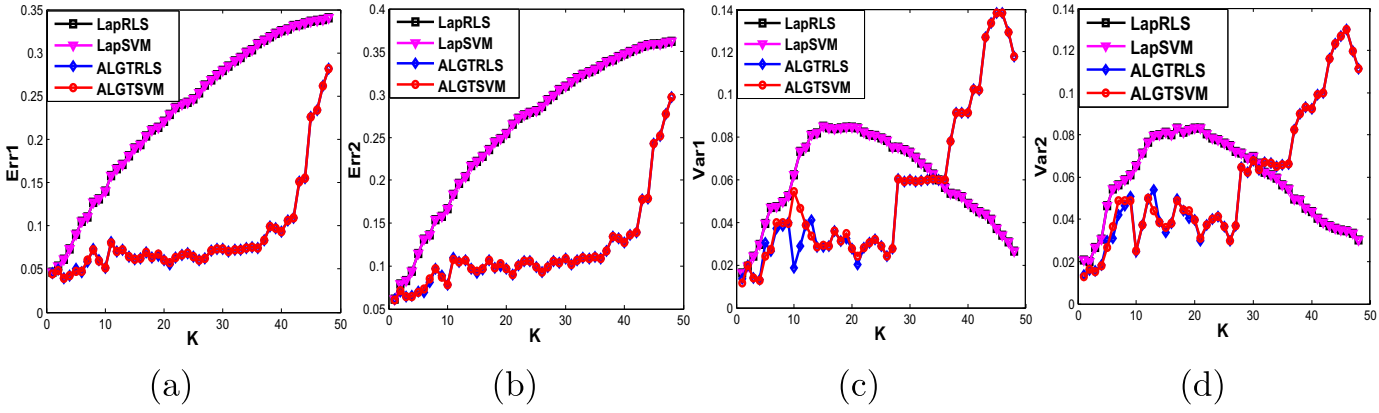

Fig. 5. Performance comparison of all competing methods on the Breast Cancer Wisconsin (Diagnostic) data set. Panels (a) and (b) show the predicted error rate Err1 and Err2 for unlabeled samples with respect to different neighborhood size K while panels (c) and (d) present corresponding variance.

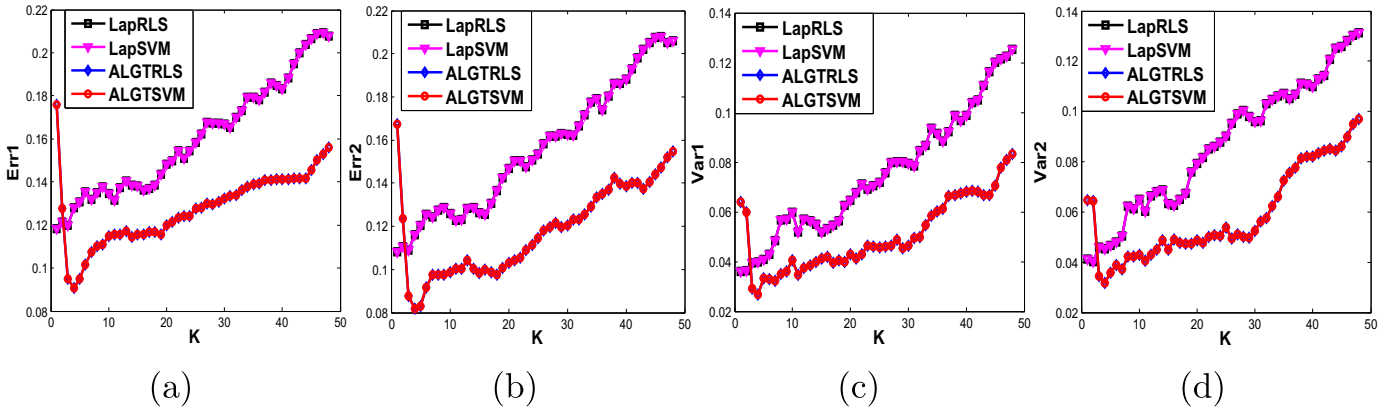

Fig. 6. Performance comparison of all competing methods on the Balance Scale data set. Panels (a) and (b) show the predicted error rate Err1 and Err2 for unlabeled samples with respect to different neighborhood size K while panels (c) and (d) present corresponding variance.

set and the test set, and y_j^{tr} and y_j^{te} represent the corresponding groundtruth ones.

There are three regularization parameters, $\gamma_A, \gamma_I, \gamma_X$ involved in the proposed model. For γ_A and γ_I, we follow the suggested specification in lapsvmp_v02 library [19] and fix $\gamma_A = 1 \times 10^{-5}$ and $\gamma_I = 1$ in our experiments. As for γ_X, we have empirically found that our method can consistently perform well when we set it in the interval $[1,2]$ throughout all our experiments (see more details in Section 4.3). We thus just simply fixed $\gamma_X = 1$ in our algorithm.

4.1. Synthetic data sets

We compare ALGT method with LapSVM and LapRLS [3,19] on two synthetic data sets, two moons data set and ellipse data set. The original simple two moons data set is available in the lapsvmp_v02 library, and we both add 200 points on previous train set and test set so as to increase the complexity of the data set as shown in Fig. 1. Another ellipse data set depicted in Fig. 2 containing 400 points is constructed by ourself, inspired by Wang et al. [28], and composes of two classes, each containing 200 samples. We randomly separate it into train set and test set, which both contain 200 samples. The data are randomly generated from the two dimensional Gaussian distribution with mean $(0,-1.3)$ and covariance matrix $\begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix}$, and the other 200 points in another class is randomly generated from the following set:

$$\left\{(x,y) \mid y = y_0 + N(0,0.1), x^2 + \frac{y_0^2}{4} = 1\right\}.$$

The first series of experiments demonstrates the phenomenon of shortcut as shown in Figs. 1 and 2, where the red dashed lines correspond to the shortcuts causing fault label information propagation within the graph. Our method can break up these shortcuts after 4 alternative iterations for the two moons data set as shown in Fig. 1 (b)–(f) and 2 alternative iterations for the ellipse data set as shown in Fig. 2 (b)–(d). This is due to the fact that the weight w_{ij} of the shortcut will decrease to zero through Eq. (7) and Eq. (8) during the learning process of our method by considering the feedback from the label knowledge on data. It is then substantiated that the shortcut issue can be effectively alleviated by our method.

The second series of experiments compares the performance of all competing methods under varying neighborhood size K. Fig. 3 shows the average error rate tendency curve with respect to K for two moons data set, and Fig. 4 shows those for ellipse data set. It can be observed that on one hand, our method always performs as accuracy as or, in more cases, more accurate, than other competing methods (in terms of both mean and variance) by adopting the ALGT technique under various K's settings; and on the other hand, LapSVM

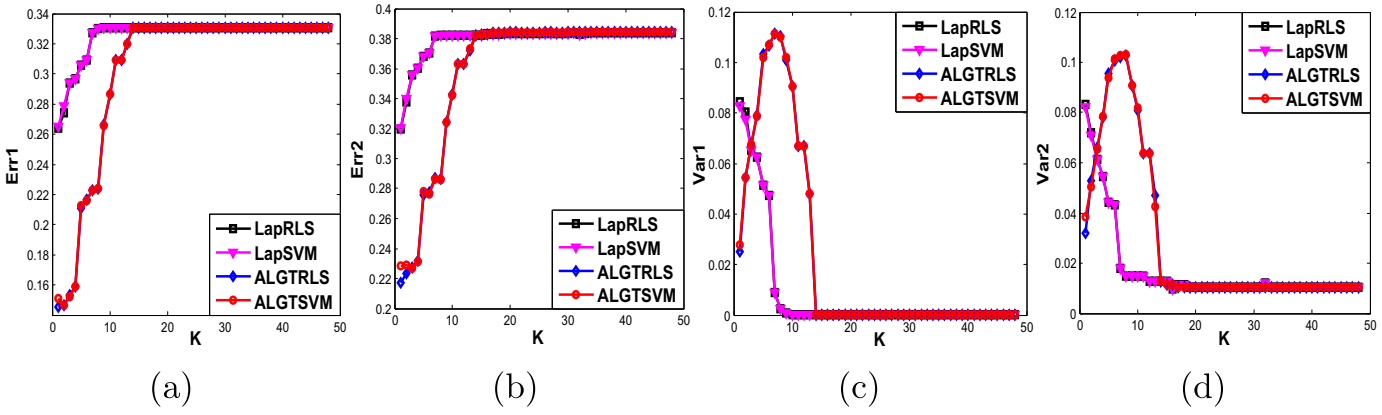

Fig. 7. Performance comparison of all competing methods on the Ionosphere data set. Panels (a) and (b) show the predicted error rate Err1 and Err2 for unlabeled samples with respect to different neighborhood size K while panels (c) and (d) present corresponding variance.

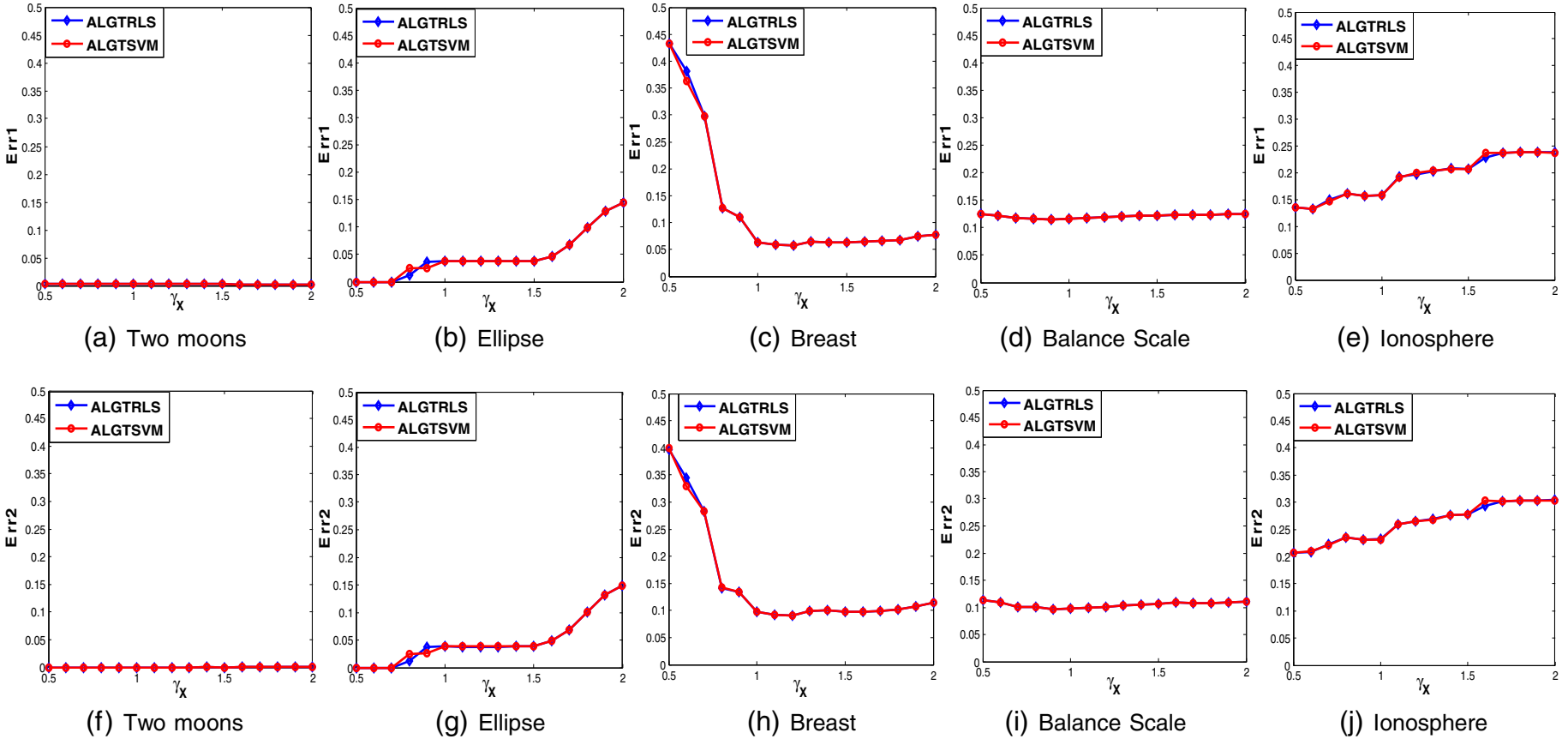

Fig. 8. This group of figures illustrates the stability of regularization parameter γ_X on five data sets. The vertical axis is the avergae Error rate Err1 or Err2 and the horizontal axis represents the γ_X ranging from 0.5 to 2.

and LapRLS are relatively more sensitive to the choice of K and error rate increases sharply with variation of K, while our ALGT method is evidently more robust to this parameter setting. This is due to the capability of the proposed method on breaking up the unexpected shortcuts, which makes it perform more stable for the selection of this important parameter.

4.2. UCI data sets

We further test ALGTSVM, AlGTRLS, LapSVM and LapRLS on three UCI data sets, including Breast Cancer Wisconsin (Diagnostic), Balance Scale and Ionosphere[2]. The numerical attributes of the data sets are all standardized. For all of the three data sets, we vary the number of nearest neighbors and then compare the mean and variance of the unlabeled samples prediction accuracy. As for the ALGT method with weight matrix in the format Eq. (8), the bandwidth is set as the maximum value between each selected samples which are the NN-samples during iteration in step 4 of Algorithm 1.

Figs. 5, 6 and 7 compare the performance of all competing methods. The vertical axis presents the average error rate Err1 and Err2 or corresponding variance computed over 20 random experiments and horizontal axis is the value of nearest neighbor number K. It is easy to see our ALGT method has lower average error rate and lower variance for most different neighborhood sizes. The ALGTSVM and ALGTRLS all demonstrate more evident robustness on selection of K than the competing methods. This substantiates the effectiveness of the proposed method on these UCI data.

It should be noted that in the Ionosphere experiments, our ALGT method has larger variance than original manifold regularization method for K ranging from 6 to 15, as shown in Fig. 7 (c) and (d). This is due to the fact that under such neighborhood size settings, other competing methods consistently perform not well (which can be observed from their evidently larger average error rates under such neighborhood size settings), while the performance of our methods is good in some samples and might be degenerated in others. It is thus still reasonable to say that the proposed method has a better performance in such cases.

4.3. Performance sensitivity test to the γ_X setting

In this section we show some experiments to evaluate how the setting of the parameter γ_X influences the performance of the proposed methods. Fig. 8 shows the tendency curves of the average error rates (Err1 and Err2), obtained by the proposed methods on all of the utilized synthetic and UCI data sets, with respect to different settings of γ_X ranging from 0.5 to 2. From the figure, it is easy to observe that our methods have a consistently stable performance especially when γ_X is in the interval $[1, 2]$. We thus simply set γ_X as 1 throughout all our experiments, and our method can consistently perform well under this setting.

5. Conclusion and discussion

The existing GSSL methods, such as LapSVM and LapRLS, heavily rely on the choice of neighborhood size, especially when data set is of a complex manifold configuration. To alleviated this issue, this work has proposed an adaptive robust Laplacian graph trimming (ALGT) semi-supervised learning (SSL) strategy, which is capable of adaptively adjusting the weights of the NN-graph imposed on data through supplementally considering the feedback knowledge of label prediction during the learning process of our method. Such an adaptive adjustment mechanism helps break up most of the unexpected connections (shortcuts) involved in the NN-graph and enhances the robustness of our method to the selection of such important parameter in GSSL problem. In addition, our method can be served as a general framework for manifold regularization methods, and is hopeful to help rectify the Laplacian regularizer in any related manifold learning tasks.

In our further work, we are interested in extending this Laplacian-regularizer-learning framework to more machine learning and computer vision tasks, like transfer learning and domain adaptation, and extracting the theoretical insight under this newly designed regularization form.

[2] http://archive.ics.uci.edu/ml/

Acknowledgments

This research was supported by the National Grand Fundamental Research 973 Program of China under Grant No. 2013CB329404, the China NSFC project under contract 61373114, and partially by Macau FDCT project with no. 019/2014/A1.

References

[1] M. Belkin, P. Niyogi, Laplacian Eigenmaps for dimensionality reduction and data representation, Neural Comput. 15 (6) (2003) 1373–1396.
[2] M. Belkin, P. Niyogi, Using manifold structure for partially labeled classification, Adv. Neural Inf. Proces. Syst. (NIPS) (2003)
[3] M. Belkin, P. Niyogi, V. Sindhwani, Manifold regularization: a geometric framework for learning from labeled and unlabeled examples, J. Mach. Learn. Res. 7 (1) (2006) 2399–2434.
[4] A. Blum, J. Lafferty, M.R. Rwebangira, R. Reddy, Semi-supervised learning using randomized mincuts, International Conference on Machine Learning (ICML), 2004.
[5] V. Castelli, T.M. Cover, On the exponential value of labeled samples, Pattern Recogn. Lett. 16 (1) (1995) 105–111.
[6] S.I. Daitch, J.A. Kelner, D.A. Spielman, Fitting a graph to vector data, International Conference on Machine Learning (ICML), 2009.
[7] F. Dornaika, A. Bosaghzadeh, H. Salmane, Y. Ruichek, Graph-based semi-supervised learning with local binary patterns for holistic object categorization, Expert Syst. Appl. 41 (17) (2014) 7744–7753.
[8] A.B. Goldberg, X. Zhu, Seeing stars when there aren't many stars: graph-based semi-supervised learning for sentiment categorization, Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing, TextGraphs-1, 2006. pp. 45–52.
[9] J.S.S. III, D. Ramanan, Self-paced learning for long-term tracking, Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, 2013.
[10] T. Jebara, J. Wang, S.-F. Chang, Graph construction and B-matching for semi–supervised learning, International Conference on Machine Learning (ICML), 2009.
[11] L. Jiang, D. Meng, T. Mitamura, A.G. Hauptmann, Easy samples first: self-paced reranking for zero-example multimedia search, ACM International Conference on Multimedia (ACM MM), 2014.
[12] L. Jiang, D. Meng, S.I. Yu, Z. Lan, S. Shan, A. Hauptmann, Self-paced learning with diversity, Neural Information Processing Systems (NIPS), 2014.
[13] L. Jiang, D. Meng, Q. Zhao, S. Shan, A. Hauptmann, Self-paced curriculum learning, AAAI Conference on Artificial Intelligence (AAAI), 2015.
[14] M.P. Kumar, B. Packer, D. Koller, Self-paced learning for latent variable models, Advances in Neural Information Processing Systems (NIPS), 2010.
[15] B.B. Liu, Z.M. Lu, Image colourisation using graph-based semi-supervised learning, IET Image Process. 3 (3) (2009) 115–120.
[16] X. Liu, S. Pan, Z. Hao, Z. Lin, Graph-based semi-supervised learning by mixed label propagation with a soft constraint, Inf. Sci. 277 (2) (2014) 327–337.
[17] M. Maier, M. Hein, U.V. Luxburg, Cluster identification in nearest-neighbor graphs, Algorithmic Learning Theory, International Conference, 2007. pp. 196–210.
[18] M. Maier, U.V. Luxburg, M. Heiny, Influence of graph construction on graph-based clustering measures, Adv. Neural Inf. Proces. Syst. (NIPS) (2008)
[19] S. Melacci, M. Belkin, Laplacian support vector machines trained in the primal, J. Mach. Learn. Res. 12 (5) (2009) 1149–1184.
[20] D. Meng, Q. Zhao, What objective does self-paced learning indeed optimize? Comput. Sci. (2015)
[21] Min, Adaptive KNN Classification Based on Laplacian Eigenmaps and Kernel Mixtures. Tech. Rep., University of Toronto. 2008.
[22] P. Niyogi, On manifold regularization, Int. Work. Artif. Intell. Stat. (AISTAT) (2005)
[23] X. Peng, H. Tang, L. Zhang, Z. Yi, A unified framework for representation-based subspace clustering of out-of-sample and large-scale data, IEEE Trans. Neural Netw. Learn. Syst. (2016) http://dx.doi.org/10.1109/TNNLS.2015.2490080.
[24] J. Ratsaby, S.S. Venkatesh, Learning from a mixture Of labeled and unlabeled examples with parametric side information, Annual Conference on Computational Learning Theory, 2003.

[25] C. Rosenberg, M. Hebert, H. Schneiderman, Semi-supervised self-training of object detection models, IEEE Workshop on Motion and Video Computing (WACV/MOTION), 2010.
[26] M. Szummer, Partially labeled classification with Markov random walks, Adv. Neural Inf. Proces. Syst. (NIPS) (2002)
[27] K. Tang, V. Ramanathan, F.F. Li, D. Koller, Shifting weights: adapting object detectors from image to video, Adv. Neural Inf. Proces. Syst. (NIPS) (2012)
[28] J. Wang, T. Jebara, S.F. Chang, Semi-supervised learning using greedy max-cut, J. Mach. Learn. Res. 14 (1) (2013) 771–800.
[29] J.L. Yong, K. Grauman, Learning the easy things first: self-paced visual category discovery, IEEE Conference on Computer Vision & Pattern Recognition (CVPR), 2011.
[30] C. Zhang, F. Wang, Graph-based semi-supervised learning, Artif. Life Robot. 30 (1) (2008) 174–179.
[31] D. Zhang, D. Meng, C. Li, L. Jiang, A self-paced multiple-instance learning framework for co-saliency detection, IEEE International Conference on Computer Vision, 2015. pp. 594–602.
[32] Q. Zhao, D. Meng, L. Jiang, Q. Xie, Z. Xu, A.G. Hauptmann, Self-Paced Learning for Matrix Factorization, 2015.
[33] D. Zhou, O. Bousquet, T.N. Lal, J. Weston, B.S. Olkopf, Learning with local and global consistency, Adv. Neural Inf. Proces. Syst. (NIPS) (2004)
[34] X. Zhu, Z. Ghahramani, J.D. Lafferty, Semi-supervised learning using Gaussian fields and harmonic functions, International Conference on Machine Learning (ICML), 2003.